ABSTRACT

Splitting process in Bounding-Volume Hierarchies (BVH) for collision detection is one of the most challenging issues in computer graphics. The splitting process requires an object with their set of triangles to be splitted into two parts using binary tree. The problem exists where most of objects do not have similar triangle size and thus creating unbalanced tree when reaching the final level of tree construction. It is very crucial to make sure that the BVH tree construction is always in balanced as the speed of BVH tree traversal algorithm dropped for unbalanced tree. In this thesis, we introduce the Spatial Object Median Splitting (SOMS) Rule to enhance the capability of BVH construction by efficiently splitting the irregular triangles so that each of them can be bounded with single Axis-Aligned Bounding Box (AABB). The process starts by splitting the longest axis between the midpoint of the triangles based on their minimum and maximum points. Result show that SOMS is capable of creating an optimum level of BVH where more leaf nodes containing a single triangle bounded with AABB are produced compared to Spatial Median technique. From the BVH construction experiments, SOMS managed to create the AABB tree in 43 milliseconds for object that have 948 triangles or 8% faster compared to the Spatial Median technique. Furthermore, experiment to create BVH also showed that SOMS produced 26% more nodes with single triangle compared to the Spatial Median technique given that the total object triangle count for the complex environment are 5672 triangles. The E-Node traversal approach that is specially designed to perform traversal testing for SOMS technique is able to detect overlap between dynamic and static models in an average of 0.2 milliseconds for the same complex environment. As a conclusion, BVH can easily be constructed using SOMS approach to create a more balanced tree and an E-Node traversal algorithm that is more efficient and faster.

ABSTRAK

Proses pemisahan di dalam Isipadu Persempadanan Berhierarki (BVH) untuk pengesanan pelanggaran adalah salah satu isu paling mencabar dalam bidang grafik berkomputer. Proses pemisahan ini memerlukan set segitiga objek dipisahkan kepada dua bahagian menggunakan pepohon binari. Masalah muncul apabila objek tersebut tidak mempunyai saiz segitiga yang sama dan membuatkan proses pemisahan membina pepohon BVH yang tidak seimbang. Kepantasan algorithma penyusuran pepohon juga menurun jika pepohon BVH tidak seimbang. Di dalam tesis ini, satu teknik yang dipanggil Peraturan Pemecahan Ruang Median berdasarkan Objek (SOMS) telah diperkenalkan untuk meningkatkan keupayaan pembinaan BVH dengan membahagikan secara efisien set segitiga supaya setiap segitiga dilitupi oleh Isipadu Persempadanan Paksi Sejajar (AABB). Proses bermula dengan memotong paksi terpanjang yang dikira antara titik tengah segitiga-segitiga berdasarkan titik minimum dan maksimum. Hasilnya adalah SOMS berupaya membina pepohon BVH yang lebih optimum di mana terdapat lebih banyak nod dedaun yang mengandungi satu segitiga yang disempadani oleh AABB dihasilkan berbanding dengan teknik Ruang Median Terbuka (Spatial Median). Ujian kepantasan pembinaan BVH menunjukkan SOMS berupaya membina pepohon AABB dalam 43 milisaat bagi objek yang mengandungi sebanyak 948 segitiga atau 8% lebih pantas daripada teknik Ruang Median Terbuka. Tambahan pula, hasil ujikaji untuk membina BVH menunjukkan SOMS dapat menghasilkan 26% lebih banyak nod yang mengandungi satu segitiga berbanding teknik Spatial Median di mana jumlah segitiga untuk membentuk persekitaran kompleks tersebut adalah sebanyak 5672 segitiga. Pendekatan susuran E-Node yang direka khas untuk teknik SOMS membolehkan penyusuran antara nod yang bersentuhan antara objek dinamik dan statik dihasilkan dalam sela masa purata 0.2 milisaat bagi persekitaran kompleks yang sama. Sebagai kesimpulan, teknik SOMS membolehkan pembinaan pepohon BVH dilakukan dengan pantas dan mampu membina peringkat pepohon BVH yang lebih seimbang di samping algorithma penyusuran E-Node yang efisien dan pantas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CG	-	Computer Graphics
3D	-	Three-Dimensional
FPS	-	Frame per-second
BVH	-	Bounding-Volume Hierarchies
CA	-	Cellular Automata
UTM	-	Universiti Teknologi Malaysia
UCLA	-	University of California, Los Angeles
LAX	-	Los Angeles International Airport
GJK	-	Gilbert-Johnson-Keerthi
CCD	-	Continuous Collision Detection
AABB	-	Axis Aligned Bounding Box
OBB	-	Oriented Bounding Box
BV	-	Bounding-Volume
DOP	-	Discrete Oriented Polytope
Min	-	Minimum
Max	-	Maximum
Ghz	-	Giga Hertz
DDR	-	Double Data Rate
RAM	-	Random Access Memory
GB	-	Gigabyte (1,000,000,000 Bytes)

JPEG	-	Joint Photographic Experts Group
MB	-	Megabyte (1,000,000 Bytes)
PLY	-	Polygon File Format
3DS	-	3D Studio Max File Format
TXT	-	Text File
DFS	-	Depth First Search
BFS	-	Breadth First Search
SOMS	-	Spatial Object Median Splitting
SKP	-	Google SketchUp 3D File Format

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The visualization of complex environments in recent years has tremendously caught attention from various types of user especially for three-dimensional (3D) applications such as games, animation and simulation. Researchers, gaming programmer, animator, or normal people will be able to experience the uniqueness of complex environments. From the beginning of 3D era, various types of technique also be implemented and tools have been developed in order to create an efficient and realistic 3D applications. For example, films such as Transformers, 2012, Dragon Wars, Star Trek, and 3D animation films such as Shrek and Toy Story have shaped the future requirement of complex environments. Computer games also have increased their level of realism to produce better complex environments complete with good story line and realistic complex environments. This also helps to encourage some of the researchers to continue producing better system in visualizing the complex environments.

There was a significant amount of researches concerning on complex environments proposed by various researchers (Bittner, Wonka et al. 2001; Willmott, Wright et al. 2001; Wonka 2001; Hamill and Carol 2003; Waddell and Borning 2004; Brosnan, Hamill et al. 2005; Himanshu, Subhrajit et al. 2006; Laycock, Ryder et al. 2007; Xiaoping, Xia et al. 2007; Batya, Alan et al. 2008; Borning, Hana et al. 2008; Haciomeroglu, Laycock et al. 2008). Most of these researchers proposed techniques to be used in Urban Simulation that represent complex environments. For instance, the urban planner such as architect can properly show their design of urban planning using 3D tools. It also helps to envision urban issues in urban economies, social and political structures and norms, transportation and other infrastructure systems and technologies.

Simulating complex environments consumes lot of resources as it needs to load multiple objects that has thousand of polygons (Wonka 2001; Hamill and Carol 2003). In order to design better system and to visualize the complex environments, several important issues must be considered by programmer or researchers. For instance in most 3D applications, there are needs to maintain the suitable frame rates such as 30 – 60 frame per seconds (FPS) [(Hamill and Carol 2003; Luiz Gonzaga da and Soraia Raupp 2006; Spillmann, Becker et al. 2007; Sulaiman, Bade et al. 2008)]. To achieve that requirement, the 3D applications should be able to perform some optimization methods that could reduce the complexity of the complex environments. Such requirements also raised other issues which there are also a need to balance between the realism and the complexity of the applications [(Cohen, Manocha et al. 1994), (Luiz Gonzaga da and Soraia Raupp 2006)].

Among optimization technique that is implemented for complex environments is collision detection. Collision detection is very essential for realistic effects in complex environments where it involves static and moving objects. For example, the meteorite effects such as in 3D films required high performance collision detection method where it is involved lot of mathematical calculation. Every trajectory of meteorite needs to be calculated and when the collision occurs, it requires sophisticated mathematical calculation just to measure the impact. The

response of such collision also needs to be calculated in order to visualize the complex environments after the impact has occurred. Thus, the computational cost is increased just to confirm the collision detection that has been inserted into complex environments produced realistic effects (Tecchia and Chrysanthou 2000; Hamill and Carol 2003; Ryder, Flack et al. 2005; Luiz Gonzaga da and Soraia Raupp 2006; Laycock, Ryder et al. 2007).

The common approach that had been used by researchers when required to perform collision detection is Bounding-Volume Hierarchies (BVH) technique (Somchaipeng, Erleben et al. 2005; Nguyen 2006; Chang, Wang et al. 2008; Sulaiman, Bade et al. 2009; Tu and Yu 2009). BVH technique provides a hierarchical representation of object that could be used for complex environments by successfully dividing the object into several parts. When performing collision detection checking, only potential colliding parts is going to be further testing for collision. BVH used various type of hierarchical volume called Bounding Volume (BV) (Zachmann 2000; Bade, Suaib et al. 2006; Nguyen 2006; Kockara 2007). BV provides a representation group of primitives reducing the need to perform earlier primitive-primitive testing. Performing BV-BV testing is faster compared to primitive-primitives testing as it only checks for certain parts of BVH that collides with another object. Therefore, the computation cost of detecting object interference for complex environments can be lowered as the complexity of the object is reduced.

1.2 Problem Background

Detecting object interference in large-scaled virtual environment such as complex environment becomes a challenge since decades ago as it requires the simulation to run smoothly at interactive frame-rates (Cohen, Manocha et al. 1994). (Hubbard 1993) proposed a method that increases the speed of pair wise test between

objects but they traded with the accuracy. By making object intersection faster, performance of the simulation will become faster than normal. However, it degrades the accuracy of object intersection. This is not always the best choice as exact contact determination always takes some time to produce collision pairs (Cohen, Lin et al. 1995).The objectives of detecting object interference is quite vague as it heavily depends on the simulation itself. For medical such as surgeon simulation, accuracy is more important than speed of the intersection as it must precisely contact the patient body but for complex environment, faster intersection is more relevant compared to accuracy. However, we still need to realize that it depends on what are the important things to consider for specific environment.

In 2000, (Tecchia and Chrysanthou 2000) suggested that in order to make realistic complex environment, simulating human movement using crowd simulation becomes important aspect to bring complex environment to 'life'. However, human crowds not only bring additional cost in the simulation but also decrease the performance of complex environment by making collision detection test slower (Tecchia and Chrysanthou, 2000). For example, complex environment may consist thousands of human crowds freely walking in different directions. If one tries to perform standard exact collision detection method between human crowds and static objects in complex environment, it is time consuming as every entity needs to be checked for collision (Tecchia and Chrysanthou, 2000). Hence, they decided not to compute the accuracy of detecting object interference in this case.

In order to tackle this problem, they used a simple and fast method such as height map to detect collision in complex city cramps with human and other 3D models (Tecchia and Chrysanthou, 2000). They introduced two different approaches which involves predicting the future or final position of the object. They reallocate the task to detect collision away from current object position or they focused on detecting object interference between moving particles and static objects only. Inter-particle collision is not performed to make sure the proposed method is fast enough. Accuracy has to be abandoned in order to make sure that complex environment run smoothly in exchange for faster intersection (Tecchia and Chrysanthou, 2000).

Later in 2003, (Hamill and Carol 2003) had presented Virtual Dublin complex environment that can be used as a prototype to test various projects such as crowd and traffic simulation, land design, and disseminated graphics studies. They implemented a Moller Ray-Triangle Intersection Algorithm for detecting collision in complex environment (Möller and Trumbore 1997). (Nurminen 2006) had suggested a 3D mobile map, called m-LOMA city. The city was designed with special mobile features that can keep the location-based information of the city. In m-LOMA, collision detection had been used to detect intersection between user moving line and the objects of the city. (Pelechano, Allbeck et al. 2007) created HiDAC system (High Density Autonomous Crowd) and they used collision detection technique to detect overlapping direction of human crowd. The proposed approach only occur in one single room between other human crowd and it allows faster intersection between hundreds of human crowd (Pelechano, Allbeck et al. 2007).

Hierarchical representation has not been the main focus in term of using them for collision detection in complex environment. From thorough discussion of collision detection in complex environment above, the method that involving hierarchical representation is on grid system and height map system to detect collision (Cohen, Manocha et al. 1994; Tecchia and Chrysanthou 2000; Laycock, Ryder et al. 2007). By using so called grid system, the collision is to be determined once the object or crowd has intersected with any grid boxes. However, this method is far from more specific approach of hierarchical representation which is Space subdivision and bounding-volume hierarchies (BVH). BVH usage is more on general collision detection in virtual environment and from the study, BVH has only very little literature in large-scaled simulation such as massive environment simulation and complex environment.

In BVH, there are generally three phase of BVH construction. First phase is to determine its characteristics such as types of the tree, bounding-volume to use, splitting point, and any other characteristics. Second phase involved the construction of defined tree after all characteristics have been decided. Last phase involved the traversal scheme when it uses to detect collision between BVH of other objects.

However, looking at BVH construction in details, most researchers used common splitting algorithm to split their node into two equal sizes especially in binary type tree which is the most common BVH tree that has been used. This common method has been used along with heuristic determination in order to reach desired stopping criteria. Spatial Median technique is the most common splitting technique after Object Median technique and any other techniques (Cormen, Leiserson et al. 1990; Lin and Manocha 2004; Spillmann, Becker et al. 2007; Kun, Qiming et al. 2008; Masatake, Yasuyuki et al. 2008). They work in different ways as Spatial Median cut the object into two equal parts while Object Median calculate median point and split at the centre of object median.

Most objects that are designed using 3D tools theoretically contain almost the same triangle size. But in practise, most 3D designer prefers to give some modification to the object in order to draw their object according to their own specification. Thus, the triangle of the corresponding object is not the same as in theory. Hence, by enclosing the object with BVH could produce the BVH to certain levels but eventually stop when it reaches non splitting condition such as irregular size of triangles. Hence researchers come out with heuristic determination in order to continue split the object into desired stopping point according to stopping criteria or until each node contain only one triangle. Among splitting that used is that kind of heuristic in their implementation are both spatial median and object median which are the most common implementation for BVH construction or more general in hierarchical representation (Rusinkiewicz and Levoy 2000; Sobottka and Weber 2005; C.Watcher and A.Keller 2006; Ernst and Greiner 2007; Liu, Wang et al. 2007; Otaduy, Chassot et al. 2007; Yi, Kim et al. 2007; Steinemann, Otaduy et al. 2008).

In this research, the splitting technique will be improved by increasing the level of BVH that can be generated automatically without using heuristic at the earlier phase. It reduces time to construct tree and increase the efficiency in collision detection handling for large complex environment. As complex environment consist lot of buildings, performing collision in large area could become problem as it could consumes memory and constantly test for object intersection. Hierarchical

representation could reduces the problem into certain degrees by making only parts of object that within the hierarchical section is tested for intersection leaving any non-colliding pairs.

In this research, our main research question is.

What kind of optimization approach that is suitable in BVH technique that will improved the capabilities of BVH construction in order to create fast and efficient technique to perform collision detection between static and moving object for complex environments.

In order to find the solution of creating faster collision detection method using BVH technique for complex environments, few questions need to be detailed up as follows:-

1. How to find the correct improvement on BVH that possibly improve the collision detection method for complex environments.
2. What type of tree construction must be used? Is it binary tree, quad tree, or oct-tree?
3. How to choose the tree construction approach? Is it top-down, bottom-up and insertion method?
4. How to choose to splitting/merging criteria for each branch of the tree hierarchies?
5. Why each splitting rule give differences result when it is used for BVH?
6. What types of bounding-volumes suitable for proposed technique?
7. Does the chosen technique speed up the process of detecting object interference? And how fast compared to the previous one and how to measure the ‘accuracy’ of the technique.

For each question described above, the answer for all research questions invoked will be answered in chapter two, three, and four.

1.3 Research Aim

The aim of this research is:-

To enhance the splitting rule technique using Binary BVH of Axis-Aligned Bounding Boxes (AABB) in detecting object interference for complex environments therefore the performance of collision detection handling can be optimized.

1.4 Research Objectives

The objectives for this research study are as follows:

1. To develop Spatial Object Median Splitting (SOMS) Rules
2. To develop E-Node Traversal Algorithm for SOMS technique
3. To evaluate and benchmark the proposed techniques with previous method of BVH splitting rules.

1.5 Scope of the study

The scope of this research is as follows:

1. Two common 3D data files which are Polygon (PLY) data file and 3D Studio Max (3DS) data file were used in this research implementation. For PLY object, three models were chosen for simple collision testing. These models can be downloaded from Stanford University Repository model library.
   a) Stanford Bunny Model
   b) Stanford Teapot Model
   c) Stanford Venus Model

2. The Urban Simulation that represents the complex environments is developed using 3D modeling tools: Google Sketchup 6.4 version. It contains all the complex environments basic requirements which are multiple objects, basic texturing and lighting.

3. The propose method is only focusing at the improvement of splitting algorithm for BVH. Other elements such as data management, culling system, texturing, and lighting will not be discussed. This is because the technique only requires the object triangle in order to find the differences between other splitting techniques.

4. An OpenGL library and C++ language will be used for the programming parts as well as implementing the Class and Data Structure concept.

5. Testing procedure will be conducted using orientation test between moving object and static object. Orientation test consists of basic directional movement of moving object either from left, right, top, and bottom towards the static object for collision detection test. The parameters involves are:-

    a) Time in milliseconds
    b) Frame
    c) Frame per second (FPS)

1.6 Significance of the study

BVH technique can be reproduced by enclosing geometry objects with AABB bounding-volumes without having to slow down the process of detecting interference in complex environments. This research intends to achieve all the objectives according to the scope of this research towards development of efficient collision detection technique for complex environments. The proposed technique is able to minimize the computation cost and increase the performance for complex environments when collision occurred. As a result, the frame-rate of complex environments increases regardless number of object being tested.

1.7 Contribution of this thesis

The contribution of this thesis consists of:-

1. New Spatial Object Median Splitting Rules in Bounding-Volume Hierarchies for Complex environments.
2. E-Node Traversal Algorithm that is embedded with the new Splitting Rules to include in BVH technique for collision detection.

1.8 Structure of the thesis

This report consists a total of six chapters. The organisation of this report has been arranged into following orders.

Chapter 1 discusses the fundamental elements of this research. It consists introduction to this research, related work, problem statement, objective and scope of the research, as well as research contribution.

Chapter 2 discusses thoroughly about literature review of this research. Discussion starts by explaining complex environments, hierarchical representation and collision detection. Various techniques or methods have been discussed here to provides enough evidence to support this research implementation.

Chapter 3 provides a research methodoly of this research. Every phase has been thoroughly discussed. All phases are well-explained starting from literature review, data collection and analysing of previous techniques and few other phases.

Chapter 4 introduced the Spatial Object Median Splitting (SOMS) Rules technique in BVH. All discussions about how the implementation of SOMS is carried out, the justification of SOMS, and related information regarding SOMS will be explained in this chapter.

Chapter 5 is the results and discussion chapter for our proposed technique in details. The chapter also will discuss in details about all testing and results were made along with the justification on each experiement. Each implementation is explained thoroughly and why the testing must be done such way.

Chapter 6 is the final chapter where the summarization of our work. The conclusion and future work is also discussed.